

# Inteligencja obliczeniowa - Generative Adversarial Network i inne modele generujące

Grzegorz Madejski

# Trochę teorii...

- *Generatywne sieci przeciwstawne* (ang. *Generative Adversarial Network*, GAN) to model głębokiego uczenia wykorzystywany do generowania multimediów (grafik, wideo, audio, tekstów).
- Model ten najpierw uczy się z dostępnych danych. Wyuczony, jest w stanie generować podobne dane, łączyć je ze sobą w dość zaskakujące sposoby.

# Trochę teorii...

- Model GAN składa się z dwóch podmodeli: generatora i dyskryminatora.
- *Generator* to model do generowania obrazu. Dostaje on na wejście wektor liczbowy, na podstawie którego generuje obrazek.
- *Dyskryminator* to model do oceniania, czy dany obrazek jest prawdziwy, czy jest fake'iem zrobionym przez generatora (klasyfikacja binarna).
- Generator i dyskryminator walczą ze sobą. Generator próbuje oszukać dyskryminatora tworząc w procesie uczenia coraz bardziej realistyczne obrazy. Dyskryminator usiłuje coraz lepiej rozpoznawać fejki.

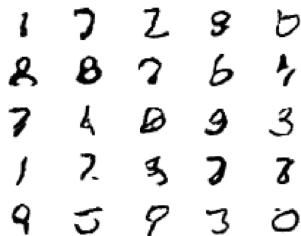
# Trochę praktyki...

Skorzystamy z samouczka

[https://machinelearningmastery.com/](https://machinelearningmastery.com/how-to-develop-a-generative-adversarial-network-for-an-mnist-dataset/)

`how-to-develop-a-generative-adversarial-network-for-an-mnist-dataset`

by nauczyć się generować obrazki przypominający ręcznie zapisane cyfry (MNIST dataset).



# Trochę praktyki...

Zademonstrujemy jak wczytać i wyświetlić obrazy z cyframi.  
Załączony plik [gan01.py](#).

# Trochę praktyki...

Zademonstrujemy jak zdefiniować dyskryminatora, jako sieć konwolucyjną. Załączony plik [gan02.py](#).

# Trochę praktyki...

Przygotujemy zestaw obrazków wygenerowanych losowo, zestawimy je z prawdziwymi obrazkami cyfr. Wyuczymy dyskryminator rozpoznawać, które są fałszywe, a które prawdziwe. Załączony plik [gan03.py](#).

# Trochę praktyki...

Definiujemy generator. Model dostaje wektor liczb (o wielkości na przykład  $latentdim = 100$ ). Dodaje warstwę ukrytą o  $126 \cdot 7 \cdot 7 = 6272$  neuronach i łączy te dwie warstwy. Robi reshape na 128 obrazków  $7 \times 7$ . Powiększa te obrazki do  $14 \times 14$ , następnie  $28 \times 28$ . A potem warstwą konwolucyjną zamienia je na wynikowy obrazek. Załączony plik [gan04.py](#).



# Trochę praktyki...

Uwaga! W `gan04.py` pojawiają się warstwy typu *conv2dtranspose*. Są to warstwy z odwróconą/transponowaną konwolucją/splotem (lub inaczej dekonwolucją, ang. deconvolution). Kernele konwolucyjne zamieniały zestaw (np. 5x5) pikseli na jeden piksel. Kernele dekonwolucyjne zamieniają jeden piksel na zestaw pikseli (np. 5x5) - czyli rozpraszają informację.

# Trochę praktyki...

Sprawdzamy jakie obrazki wygeneruje generator. Plik: [gan05.py](#).

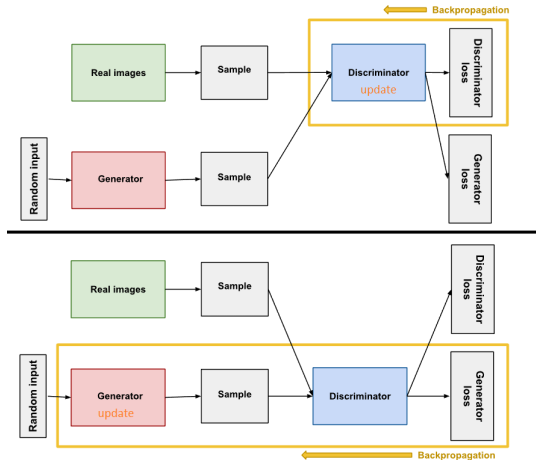
# Trochę praktyki...

Sklejamy generatora i dyskryminatora w jeden model GAN. Plik:  
*gan06.py*.

## Trochę praktyki...

Pora na ostatni krok: trening całego modelu. Na każdym batchu najpierw trenujemy sam dyskryminator (obliczając  $d\_loss$  i aktualizując jego wagi). Następnie na tym batchu trenujemy generator (mimo że propoagacja błędu przechodzi też przez dyskryminator), obliczając  $g\_loss$ . Cały model działa dobrze, gdy sprytny generator się tak wyuczy, że dyskryminator nie rozpoznaje fejków od realnych, czyli ma dokładność 50%.

# Trochę praktyki...



# Trochę praktyki...

Trenujemy GAN. Plik: [gan07.py](#). Czy wygenerowane przez niego obrazki przypominają liczby?

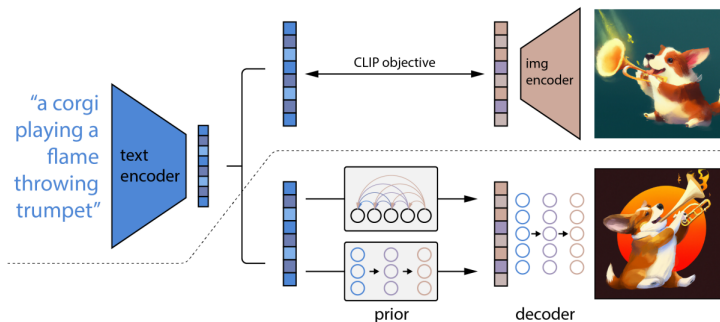
# GANPaint

Przetestujmy narzędzie GANPaint

- Narzędzie: <https://ganpaint-v2.vizhub.ai/>
- Informacje o narzędziu:  
<https://gandissect.csail.mit.edu/>

# DALL-E

*DALL-E* - narzędzie do generowania obrazów na podstawie opisu tekstowego.





# DALL-E

Przetestujmy:

- Narzędzie: <https://openai.com/dall-e-3>  
<https://www.bing.com/images/create>
- Informacje o narzędziu:  
<https://cdn.openai.com/papers/dall-e-2.pdf>  
<https://cdn.openai.com/papers/dall-e-3.pdf>

# Stable Diffusion

Przetestujmy:

- Narzędzie: <https://stablediffusionweb.com/>
- Informacje o narzędziu: <https://stability.ai/blog/stable-diffusion-public-release>
- Stable Diffusion jest modelem open-source. Wiele stron wykorzystuje go, robi fine-tuning i oferuje pół płatne narzędzia.

# Midjourney

Linki:

- Narzędzie discordowe: <https://midjourney.com/>
- Info: <https://docs.midjourney.com/>
- Przykłady wygenerowanych obrazków:  
<https://www.midjourney.com/showcase>

# Dalle3 vs Midjourney vs StableXL

Mój test dla prompta "photorealism, an old wizard with a beard and pipe, sitting in an office, drinking bubble tea and programming on a computer".

**StableXL**



**Dall-e 3**



**Midjourney**



# OpenAI Sora

- Rok 2024 może być rokiem rewolucji w generowaniu wideo.
- Wielkimi krokami nadchodzi model Sora:  
<https://openai.com/sora>